



Katalon Studio

Best Practices





The best practices described in this article provide a path to successful test automation implementation. Katalon Studio includes a number of features that help you follow these best practices.

The article also includes some essential strategies to keep in mind while performing test automation on your project.

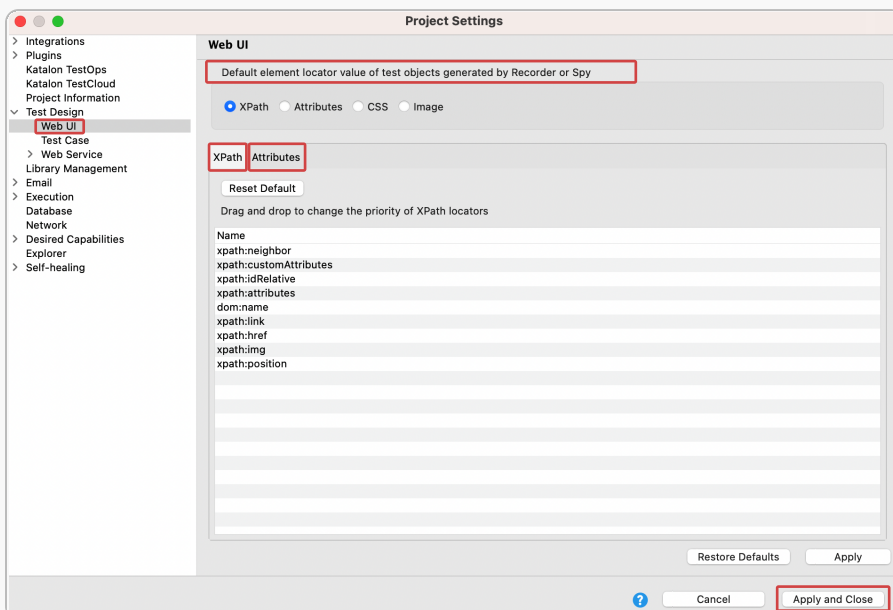
Let's begin.

Design Patterns

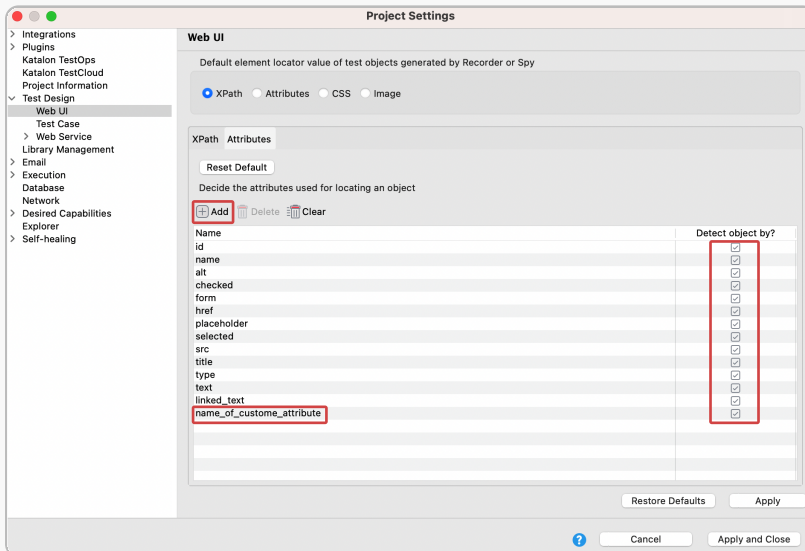
It is mandatory to create tests using good practices and patterns so that people can understand your code with minimum effort.

Learn to use record-and-playback test creation

Test recording is the easiest way for new automation testers to learn test automation. Katalon Web Recorder Utility function captures your actions performed on the application and converts them into runnable code in the back end. During record and playback, **Katalon Studio** generates an object repository and test case where we can follow the best practices.



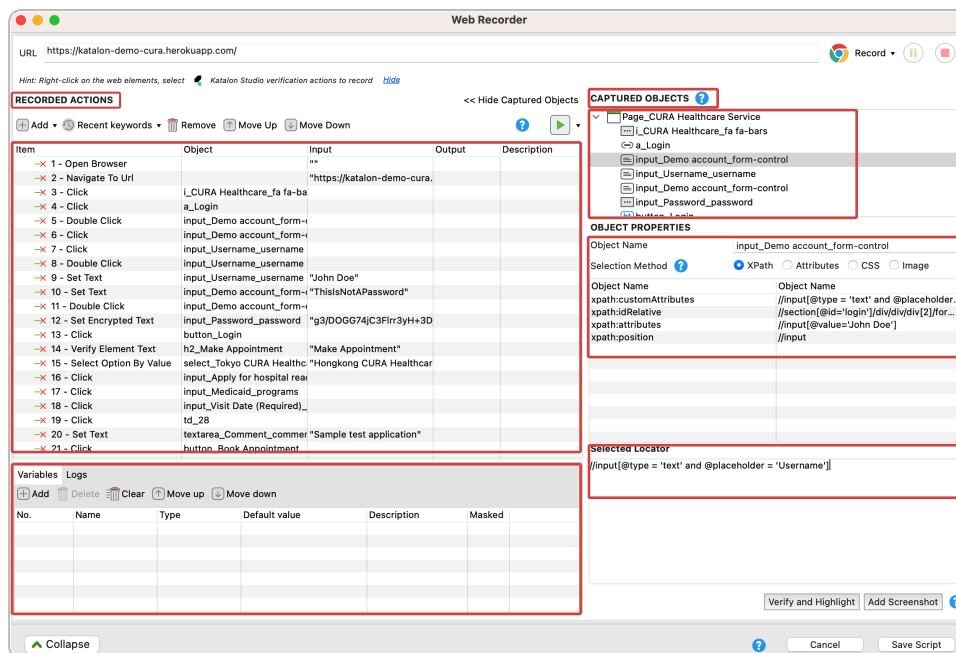
Note (optional): Before starting the record and playback, we strongly recommend users play around with the AUT DOM structure. If AUT DOM structures have dynamic properties (DOM tag values), please adjust the test design from Katalon Studio >> Project >> Settings >> Test Design >> Web UI. Users have the option to select the default element locator for Katalon Recorder and Spy.



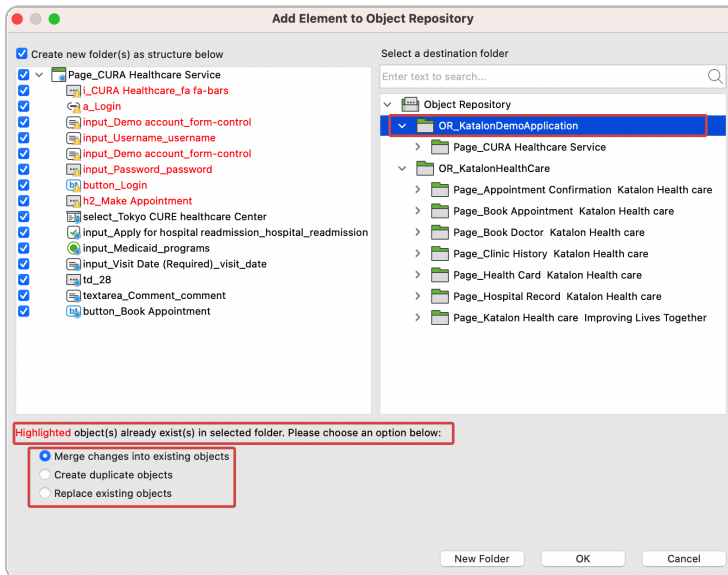
As per the AUT DOM structure/behavior users can change the priority of the XPath Locators.

If there are some specific attributes available in DOM for your AUT the user can add that attribute in the attribute section and the user can check/uncheck “Detect object by” accordingly.

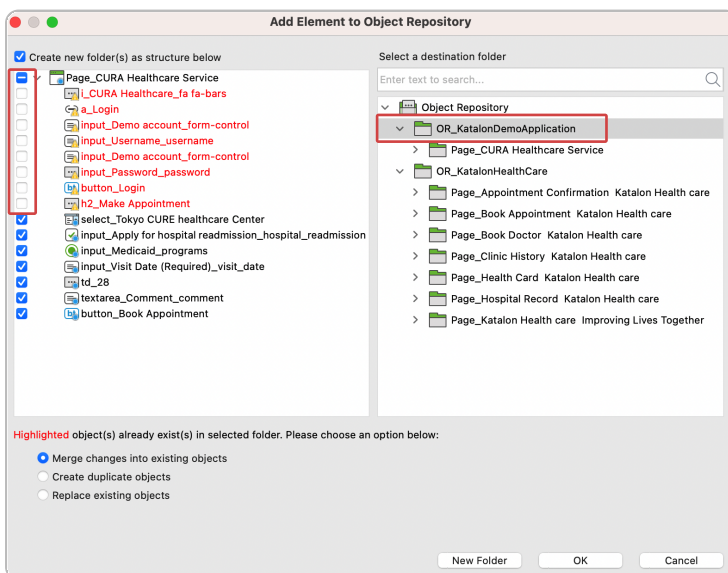
- Choose the Katalon Recorder as per application type:



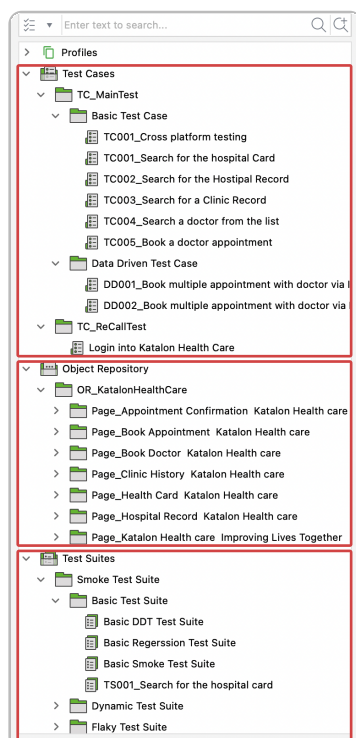
- It's always better to record the UAT test on the active browser.
- Keep verifying recorded actions by clicking on the “Expand” option for Katalon Recorder.
- Make some time to review RECORDED ACTIONS and CAPTURED OBJECTS and, if required, make some additions and deletions.
- Users can add the description for the test steps while recording or the user can add it later.
- For the CAPTURED OBJECTS do not change the name of the recorded folders and recorded objects otherwise Katalon Studio creates a new folder and new Object for every recording and that can create duplicity within the Object Repository.
- Users have the option to Pause & Resume the Recorder if necessary.
- Katalon provides some advanced actions for mouse right-click events, so use it while recording the test if required.



- It's always good to create a folder inside the Object Repository as per the module.
- First-time recording users can save all the objects with the default configurations in the module folder.
- If the user records the test again, it's always better to select the module folder first, then the Katalon Recorder will show all the duplicate objects; we suggest you choose the option to "Merge changes into existing objects" and save.



- It is recommended the user uncheck all the duplicate objects from the highlighted list and then save.

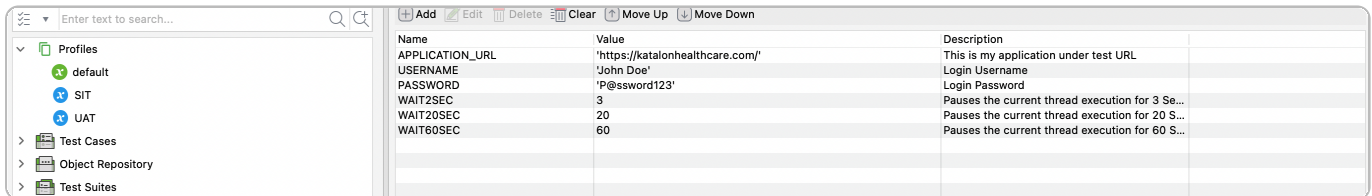


- For the Object Repository, Katalon Studio automatically follows the Page Object Model so it is always good practice to follow any folder structure for Test Case and Test Suite as well.

Naming Convention

It's always a good practice for your company/project/team to set a few rules that you need to follow for a clean code while creating a test automation framework. For example:

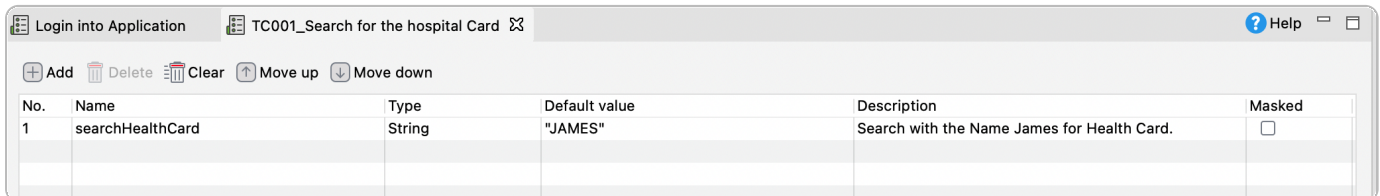
- Use uppercase string while declaring the global variable.



The screenshot shows the 'Global Variables' table in Katalon Studio. The table has columns for Name, Value, and Description. It lists several predefined variables like APPLICATION_URL, USERNAME, PASSWORD, and WAIT times.

Name	Value	Description
APPLICATION_URL	'https://katalonhealthcare.com/'	This is my application under test URL
USERNAME	'John Doe'	Login Username
PASSWORD	'P@ssword123'	Login Password
WAIT2SEC	3	Pauses the current thread execution for 3 Se...
WAIT20SEC	20	Pauses the current thread execution for 20 S...
WAIT60SEC	60	Pauses the current thread execution for 60 S...

- Use CamelCase while declaring the local/testcases variable.



The screenshot shows the 'Local Variables' table for a test case named 'TC001_Search for the hospital Card'. It has columns for No., Name, Type, Default value, Description, and Masked. One variable 'searchHealthCard' is listed with a String type and a default value of 'JAMES'.

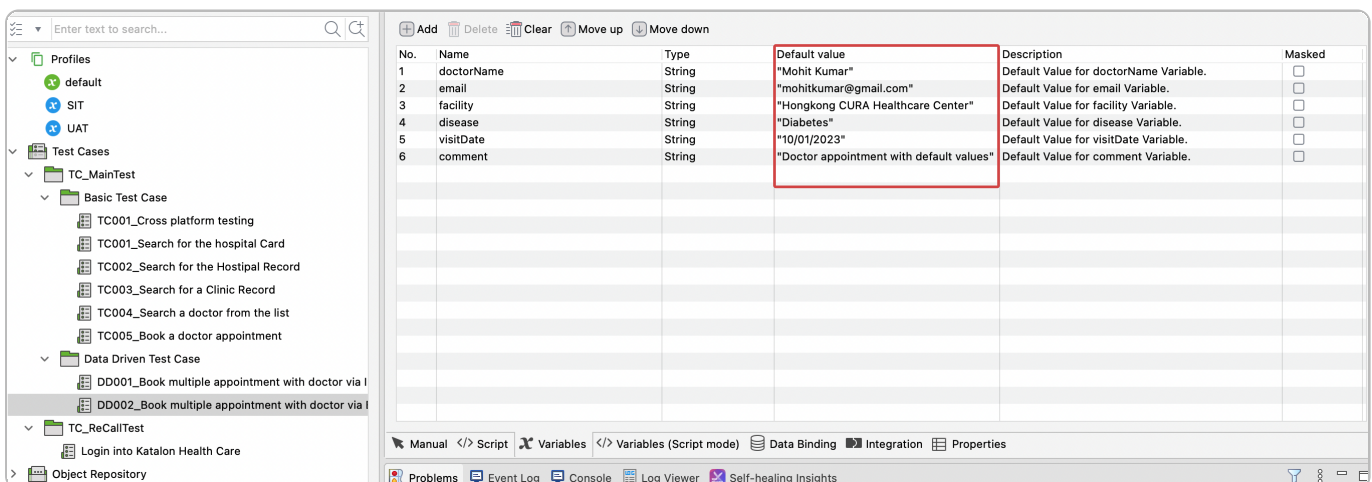
No.	Name	Type	Default value	Description	Masked
1	searchHealthCard	String	"JAMES"	Search with the Name James for Health Card.	<input type="checkbox"/>

- Avoid No meaning of variable names.
- Avoid keywords as variable names.
- Add appropriate comments/description.

Default Values and Descriptions

Always assign some default values to the variables in your test automation code.

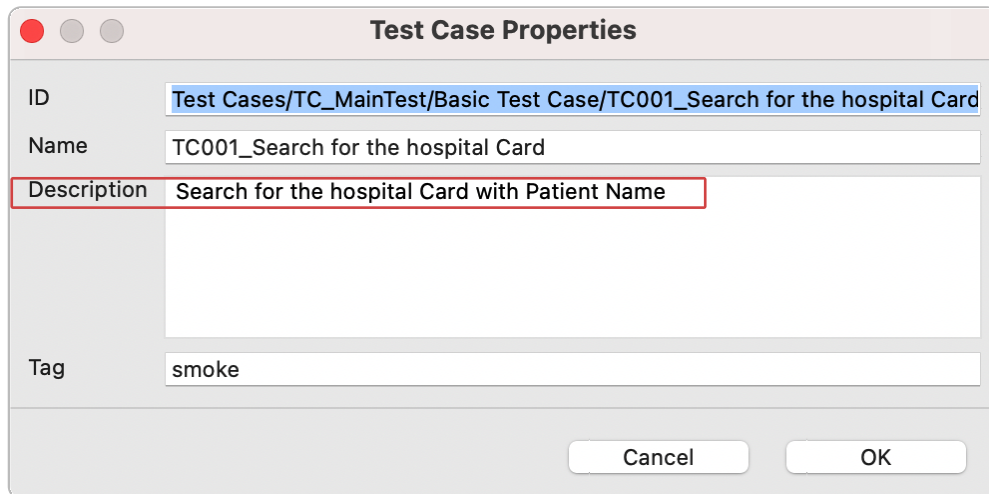
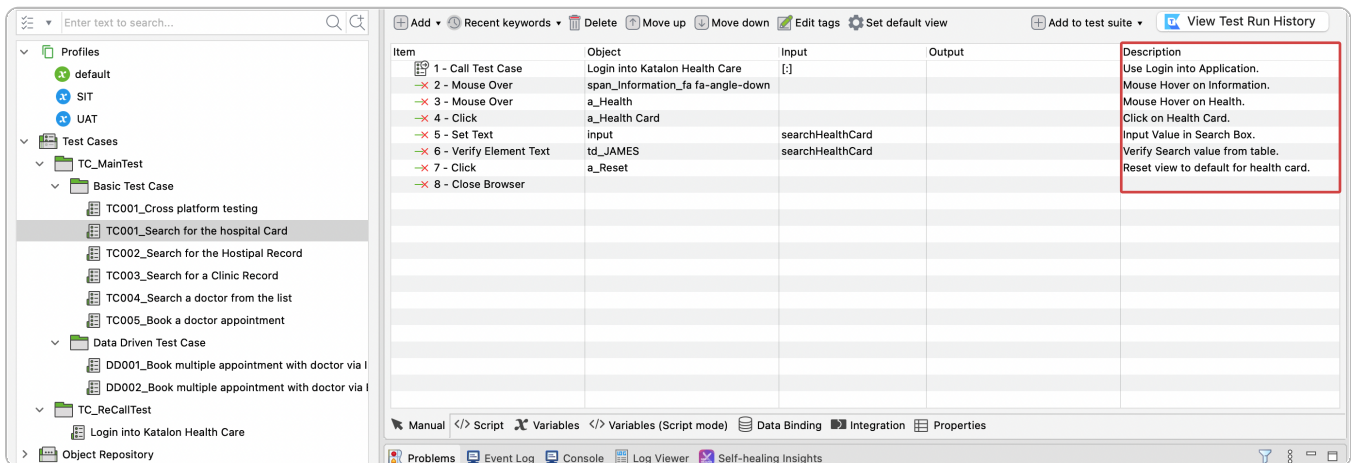
- It's always a good practice to add the default values for global variable/local variable/ data files variables, etc.



The screenshot shows the 'Local Variables' table for a test case. A red box highlights the 'Default value' column. The table lists variables like doctorName, email, facility, disease, visitDate, and comment, each with a specific default value and a description.

No.	Name	Type	Default value	Description	Masked
1	doctorName	String	"Mohit Kumar"	Default Value for doctorName Variable.	<input type="checkbox"/>
2	email	String	"mohitkumar@gmail.com"	Default Value for email Variable.	<input type="checkbox"/>
3	facility	String	"Hongkong CURA Healthcare Center"	Default Value for facility Variable.	<input type="checkbox"/>
4	disease	String	"Diabetes"	Default Value for disease Variable.	<input type="checkbox"/>
5	visitDate	String	"10/01/2023"	Default Value for visitDate Variable.	<input type="checkbox"/>
6	comment	String	"Doctor appointment with default values"	Default Value for comment Variable.	<input type="checkbox"/>

- Always add a description for test case test/test suite and for all the variables.



Do Not Hard Code Input Data or Values

Do not use any hard coding values for test cases because hard-coded input values are very dangerous. Always use some files, Excel, and a database to manage your input values.

Create Good, Quality Test Data

Test data should be meaningful and it should be as close to the user's actions and behavior as possible.

Build a test that can run the entire operating systems (OS).

```

41
42 WebUI.enhancedClick(findTestObject('Object Repository/OR_KatalonHealthCare/Page_Book Appointment Katalon Health care/label_Apply for hospital
43
44 WebUI.click(findTestObject('Object Repository/OR_KatalonHealthCare/Page_Book Appointment Katalon Health care/label_Medicaid'))
45
46 WebUI.selectOptionByLabel(findTestObject('Object Repository/OR_KatalonHealthCare/Page_Book Appointment Katalon Health care/select_MalariaHepa
47 disease, true)
48
49 WebUI.sendKeys(findTestObject('Object Repository/OR_KatalonHealthCare/Page_Book Appointment Katalon Health care/input__ninja-forms-field nf-e
50 Keys.chord(Keys.CONTROL, 'a'))
51
52 if(System.getProperty("os.name").contains("Mac"))
53 {
54     WebUI.sendKeys(findTestObject('Object Repository/OR_KatalonHealthCare/Page_Book Appointment Katalon Health care/input__ninja-forms-field
55 Keys.chord(Keys.COMMAND, 'a'))
56 }
57
58 else if(System.getProperty("os.name").contains("Windows")) {
59     WebUI.sendKeys(findTestObject('Object Repository/OR_KatalonHealthCare/Page_Book Appointment Katalon Health care/input__ninja-forms-field
60 Keys.chord(Keys.CONTROL, 'a'))
61 }
62
63 WebUI.sendKeys(findTestObject('Object Repository/OR_KatalonHealthCare/Page_Book Appointment Katalon Health care/input__ninja-forms-field nf-e
64 visitDate)
65
66 WebUI.setText(findTestObject('Object Repository/OR_KatalonHealthCare/Page_Book Appointment Katalon Health care/textarea_Comment_nf-field-65')
67 comment)
68
69 WebUI.click(findTestObject('Object Repository/OR_KatalonHealthCare/Page_Book Appointment Katalon Health care/a_Submit'))
70
71 WebUI.waitForElementPresent(findTestObject('Object Repository/OR_KatalonHealthCare/Page_Appointment Confirmation Katalon Health care/h3_Appoi
72 GlobalVariable.WAIT20SEC)
73
74 WebUI.verifyElementText(findTestObject('Object Repository/OR_KatalonHealthCare/Page_Appointment Confirmation Katalon Health care/p_Please be
75 'Please be informed that your appointment has been confirmed.')
76
77 WebUI.closeBrowser()
78
79

```

Data-Driven Testing

Data-driven testing (DDT) is the best testing framework introduced by Katalon Studio. It basically means that the input variables, data, or parameters are fetched from separate files like CSV, Excel, database, and internal data sources rather than hard coding into the same test cases.

Katalon Studio supports data-driven testing at the test case level and suite level.

DDT at the test case level: This can be very useful when users have a requirement to run the test from start to finish for all the iterations.

<div> <div> <div>+</div> <div>Add</div> </div> <div> <div>🕒</div> <div>Recent keywords</div> </div> <div> <div>🗑️</div> <div>Delete</div> </div> <div> <div>⬆️</div> <div>Move up</div> </div> <div> <div>⬇️</div> <div>Move down</div> </div> <div> <div>✏️</div> <div>Edit tags</div> </div> <div> <div>⚙️</div> <div>Set default view</div> </div> <div> <div>+</div> <div>Add to test suite</div> </div> <div> <div>📋</div> <div>View Test Run History</div> </div> </div>				
Item	Object	Input	Output	Description
1 - Call Test Case	Login into Katalon Health Care	[:]		
2 - Mouse Over	span_Services_fa fa-angle-down			
3 - Click	a_Book Doctor			
4 - Click	a_New Appointment			
5 - Click	h3_New Appointment Form			
6 - Wait For Element Present	h3_New Appointment Form	GlobalVariable.WAIT20SEC		
7 - Set Text	input_Doctor Name_fname	doctorName		
8 - Set Text	input_email	email		
9 - Select Option By Label	select_Tokyo CURA Healthcare Cente	facility; true		
10 - Enhanced Click	label_Apply for hospital readmission			
11 - Click	label_Medicaid			
12 - Select Option By Label	select_MalariaHepatitisDiabetesTub	disease; true		
13 - Send Keys	input__ninja-forms-field nf-element d	Keys.chord(Keys.CONTROL, "a")		
14 - If Statement		System.getProperty("os.name").cont		
15 - Else If Statement		System.getProperty("os.name").cont		
16 - Send Keys	input__ninja-forms-field nf-element d	visitDate		
17 - Set Text	textarea_Comment_nf-field-65	comment		
18 - Click	a_Submit			
19 - Wait For Element Present	h3_Appointment Confirmation	GlobalVariable.WAIT20SEC		
20 - Verify Element Text	p_Please be informed that your appoi	"Please be informed that your appoi		
21 - Close Browser				

Data binding:

The screenshot shows the Katalon Studio interface with the 'Test Data' tab selected. The 'Test Data' table has one row with ID '1' and data source 'Data Files/InternalDF_DoctorAppointment'. The 'Variable Binding' table below it lists six variables: doctorName, email, facility, disease, visitDate, and comment, each mapped to a specific data source.

No.	ID	Data Iteration
1	Data Files/InternalDF_DoctorAppointment	All

No.	Name	Default value	Type	Test Data	Value
1	doctorName	'Mohit Kumar'	Data Column	1 - Data Files/InternalDF_DoctorAppointment	doctorName
2	email	'mohitkumar@gmail.cc'	Data Column	1 - Data Files/InternalDF_DoctorAppointment	email
3	facility	'Hongkong CURA Heal	Data Column	1 - Data Files/InternalDF_DoctorAppointment	facility
4	disease	'Diabetes'	Data Column	1 - Data Files/InternalDF_DoctorAppointment	disease
5	visitDate	'10/01/2023'	Data Column	1 - Data Files/InternalDF_DoctorAppointment	visitDate
6	comment	'Doctor appointment v	Data Column	1 - Data Files/InternalDF_DoctorAppointment	comment

DDT at the test suite level: This can be very useful when you have a requirement to test a specific test multiple times for the attached data set.

The screenshot shows the Katalon Studio interface with the 'Execution Information' tab selected. The 'Execution Information' table has one row with ID '1' and description 'DD001_Book multiple appointment with'. The 'Variable Binding' table below it lists six variables: doctorName, email, facility, disease, visitDate, and comment, each mapped to a specific data source.

No.	ID	Description	Run
1	DD001_Book multiple appointment with	Data Driver test fo	<input checked="" type="checkbox"/>

No.	Name	Default value	Type	Test Data
1	doctorName	'Mohit Kumar'	Default	
2	email	'mohitkumar@gmail.com'	Default	
3	facility	'Hongkong CURA Heal'	Default	
4	disease	'Diabetes'	Default	
5	visitDate	'10/01/2023'	Default	
6	comment	'Doctor appointment v	Default	

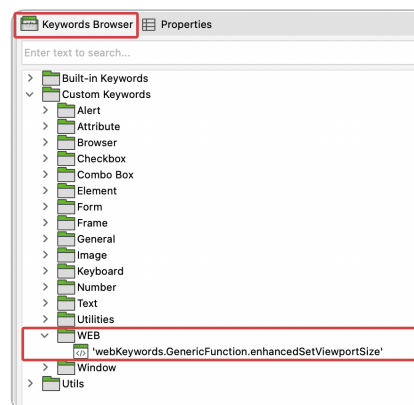
Create Custom Keywords/Wrappers

In addition to built-in keywords, users can define custom keywords to extend the capabilities of Katalon Studio. Once created, custom keywords can be used when implementing test cases, just like other built-in keywords.

We suggest our users create a separate section in “Keywords Browser” for your custom keywords via “keywordObject.”

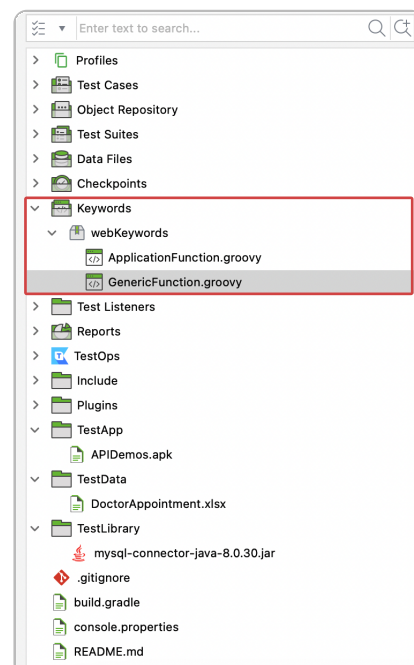
```
1 package webKeywords
2 import org.openqa.selenium.WebDriver
3
4
5
6
7
8
9
10 class GenericFunction {
11
12     @Keyword(keywordObject='WEB')
13     def static enhancedSetViewportSize(int width, int height) {
14         WebDriver webDriver = DriverFactory.getWebDriver()
15         int browserWidthGap = webDriver.manage().window().getSize().width - Integer.parseInt(WebUiBuiltInKeywords.executeJavaScript('return (w
16         int browserHeightGap = webDriver.manage().window().getSize().height - Integer.parseInt(WebUiBuiltInKeywords.executeJavaScript('return (w
17         float ratio = Float.parseFloat(WebUiBuiltInKeywords.executeJavaScript('return (window.devicePixelRatio || 1)', null).toString())
18         int actualWidth = Math.round((width + browserWidthGap * ratio) / ratio)
19         int actualHeight = Math.round((height + browserHeightGap * ratio) / ratio)
20         WebUiBuiltInKeywords.setViewportSize(actualWidth, actualHeight)
21     }
22 }
```

Keywords browser:



Generic keywords/wrappers: These can be used for all your projects within the organization, for example, to generate a random number, etc.

Application-related keywords/wrappers: These can be specific to your application only, for example, advanced search for user test applications, etc.



Test Run Notification

For a successful automation framework it's always necessary to include alerts, email reports of failure, and flaky test cases. Katalon Studio sends notification emails to help you keep track of scheduled test executions. You can receive notification emails after test execution.

Debugging the Test Case

In Katalon Studio, there are several ways to check for the reasons of failure.

Option 1: Investigate error logs

The screenshot shows the Katalon Studio interface with the Log Viewer open. The test case 'Test Cases/TC_MainTest/Basic Test Case/TC001_Search for the hospital Card' is selected. The log shows the test execution details, including the test case name, the test case ID, and the test case status. The log also shows the test case execution steps, including 'enableSauceLabs', 'callTestCase', 'openBrowser', 'navigateToUrl', 'comment', 'getTitle', 'println', and 'verifyMatch'. The log indicates that the test case failed due to a 'StepFailedException' with the message 'Actual text 'katalonhealthcare.com | 521: Web server is down' and expected text 'Katalon Health care - Improving Lives Together' are not matched'. The log also shows the test case execution time, which is 5.583s.

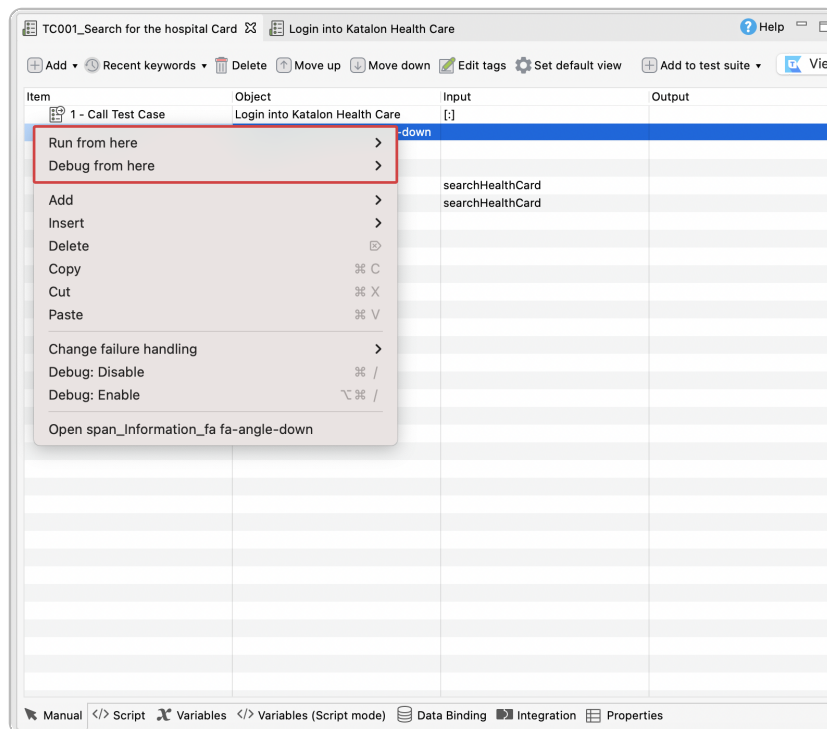
Option 2: Debug mode

The screenshot shows the Katalon Studio interface with the Script Editor open. The script is for the test case 'TC001_Search for the hospital Card'. The script includes the following steps:

- 1. Import static com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint()
- 2. Open the browser
- 3. Maximize the window
- 4. Launch the application on browser
- 5. Add url in report
- 6. Extract application title
- 7. Print the application title
- 8. Verify application title

The Console window shows the test execution logs, including the test case name, the test case ID, and the test case status. The log also shows the test case execution steps, including 'enableSauceLabs', 'callTestCase', 'openBrowser', 'navigateToUrl', 'comment', 'getTitle', 'println', and 'verifyMatch'. The log indicates that the test case failed due to a 'StepFailedException' with the message 'Actual text 'katalonhealthcare.com | 521: Web server is down' and expected text 'Katalon Health care - Improving Lives Together' are not matched'. The log also shows the test case execution time, which is 5.583s.

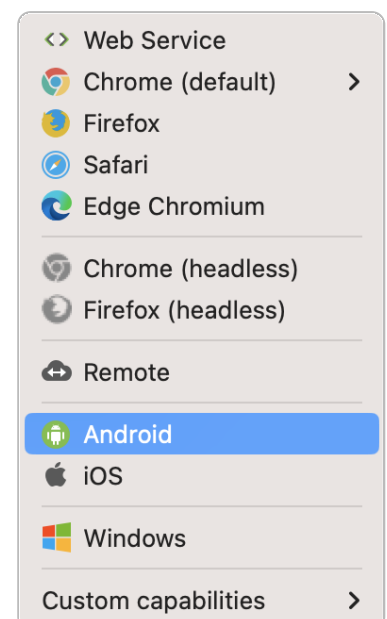
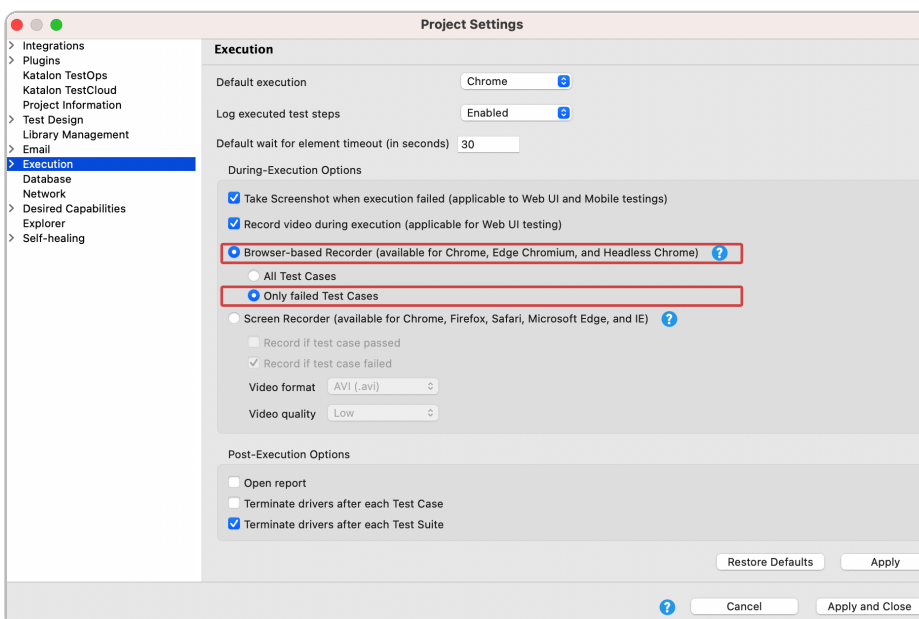
Option 3: Run/Debug from here



Test Execution

Most of the time, test cases fail due to some environmental issues or configuration issues. You need to make sure that the test cases are as stable as possible. If there are flaky tests, you should move them to a different test suite.

It's always good to keep running tests in the headless mode for smooth CI/CD execution, and use the browser-based recording with only failed test configuration.



Test Report

If you are developing a test automation, screenshots and logs are used for debugging purposes and test automation reports are best to share automation progress with the senior leadership or product owners (e.g., implement logs, screenshots, reports).

Katalon Studio generates reports only for the test suite and test suite collection; however, Katalon Studio has its own QA orchestration which delivers a very advanced capability for the test reporting.

The screenshot displays the Katalon Studio Test Report interface. The left sidebar shows a tree view of the project structure, with the 'Reports' folder expanded. The main area is divided into three panels:

- Test Cases Table:** A table with columns 'No.', 'Name', and 'Resolution'. The first row is highlighted with a red box, showing '1' in the 'No.' column and 'DD001_Book multip... Date' in the 'Name' column.
- Test Case's Log:** A log view showing the execution steps of the selected test case. The log includes a search bar and a table with columns 'Item', 'Description', and 'Elapsed'. The third item is highlighted with a red box, showing '3. click(findTestObject("Object Repository/C...))' and 'Unable to click on object 'Object Repository/C...''.
- Summary/Execution Settings/Execution Environment:** A tabbed view showing the test suite details. The 'Summary' tab is active, displaying the following information:
 - Test Suite ID: [Test Suites/Smoke Test Suite/Basic Test Suite/Basic Test Suite](#)
 - Host name: mohit - 192.168.1.10: Local OS: Mac OS X 64bit
 - Katalon version: 8.5.1.208 Platform: Chrome 105.0.0.0
 - Start: 2022-10-10 18:01:26 End: 2022-10-10 18:01:26
 - Elapsed: 23.027s
 - Total TC: 1



We hope you enjoy your testing experience with Katalon Studio.
We appreciate your feedback.